Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence

Agent-Human Coordination with Communication Costs under Uncertainty*

Asaf Frieder[1], **Raz Lin**[1] **and Sarit Kraus**[1,2]
[1] Department of Computer Science
Bar-Ilan University, Ramat-Gan, Israel 52900
[2] Institute for Advanced Computer Studies
University of Maryland, College Park, MD 20742 USA
asaffrr@gmail.com, {linraz,sarit}@cs.biu.ac.il

Abstract

Coordination in mixed agent-human environments is an important, yet not a simple, problem. Little attention has been given to the issues raised in teams that consist of both computerized agents and people. In such situations different considerations are in order, as people tend to make mistakes and they are affected by cognitive, social and cultural factors. In this paper we present a novel agent designed to proficiently coordinate with a human counterpart. The agent uses a neural network model that is based on a pre-existing knowledge base which allows it to achieve an efficient modeling of a human's decisions and predict their behavior. A novel communication mechanism which takes into account the expected effect of communication on the other member will allow communication costs to be minimized. In extensive simulations involving more than 200 people we investigated our approach and showed that our agent achieves better coordination when involved, compared to settings in which only humans or another state-of-the-art agent are involved.

Introduction

As agent technology becomes increasingly more prevalent, agents are deployed in mixed agent-human environments and are expected to interact efficiently with people. Such settings may include uncertainty and incomplete information. Communication, which can be costly, might be available for the parties to assist in obtaining more information in order to build a good model of the world. Efficient coordination between agents and people is the key component for turning their interaction into a successful one, rather than a futile one. The importance of coordination between agents and people only increases in real life situations, in which uncertainty and incomplete information exist (Woods et al. 2004). For example, Bradshaw *et al.* (2003) report on the problems and challenges of the collaboration of humans and agents on-board the international space station. Urban search-and-rescue tasks pose similar difficulties, revealed, for example, in the interaction between robots and humans during the search and rescue operations conducted at the World Trade Center on September 11, 2001 (Casper and Murphy 2003).

Teamwork has been the focus of abundant research in the multi-agent community. However, while research has focused on decision theoretic framework, communication strategies and multi-agent policies (e.g., (Roth, Simmons, and Veloso 2006)), only some focus has been on the issues raised when people are involved as part of the team (van Wissen et al. 2012). In such situations different considerations are in order, as people tend to make mistakes and they are affected by cognitive, social and cultural factors (Lax and Sebenius 1992). In this paper we focus on teamwork between an agent and a human counterpart and present a novel agent that has been shown to be proficient in such settings.

Our work focuses on efficient coordination between agents and people with communication costs and uncertainty. We model the problem using DEC-POMDPs (Decentralized Partially Observable Markov Decision Process) (Bernstein et al. 2002). The problem involves coordination between a human and an automated agent, having a joint reward (g[illegible]), while each has only partial observations of the state of the world. Thus, even if information exists, it only provides partial support as to the state of the world, making it difficult to construct a reliable view of the world without coordinating with each other.

While there are studies that focus on DEC-POMDPs, most of them pursue the theoretical aspects of the multi-agent facet but do not deal with the fact that people can be part of the team (Doshi and Gmytrasiewicz 2009; Roth, Simmons, and Veloso 2006). Our novelty lies in introducing an agent capable of successfully interacting with a human counterpart in such settings. The agent is adaptable to the environment and people's behavior, and is able to decide, in a sophisticated manner, which information to communicate to the other team member, based on the communication cost and the possible effects of this information on its counterpart's behavior.

More than 200 people participated in our experiments in which they were either matched with each other or with automated agents. Our results demonstrate that a better score is achieved when our agent is involved, as compared to when only people or another state-of-the-art agent (Roth, Simmons, and Veloso 2006) that was designed to coordinate well with multi-agent teams are involved. Our results

*This work is supported in part by ERC grant #267523, MURI grant number W911NF-08-1-0144 and MOST #3-6797.

also demonstrate the importance of incorporating a proficient model of the counterpart's actions into the design of the agent's strategy.

Related Work

In recent years several aspects of human-agent cooperation have been investigated. For example, the KAoS HART is a widely used platform for regulating and coordinating mixed human-agent teams, where a team leader assigns tasks to agents and the agent performs the action autonomously (Bradshaw et al. 2008). While in KAoS HART the agent is not performing any actions, Kamar *et al.* (2009) described settings in which an agent proactively asks for information and they tried to estimate the cost of interrupting other human team members. Rosenthal *et al.* (2010) described an agent that receives tasks, and if it expects to fail, it can ask for information or delegate sub-tasks. Sarne and Grosz (2007) reason about the value of the information that may be obtained by interacting with the user. Many of the aforementioned approaches do not consider how their actions may conflict with the actions of other team members. In a different context, Shah *et al.* (2011) showed that the coordination of a mixed human-agent team can improve if an agent schedules its own actions rather than waiting for orders. Unlike our approach, their agent does not employ a model to predict the human behavior, but it can adapt if the human partner deviates from optimal behavior. In addition, they are more concerned with timing coordination than with action coordination.

Zuckerman *et al.* (2011) improved coordination with humans using focal points. Breazeal *et al.* (2008) showed how mimicking body language can be used by a robot to help humans predict the robot's behavior. Broz *et al.* (2008) studied the POMDP model of human behavior based on human-human interaction and used it to predict and adapt to human behavior in environments without communication. We, however, focus on the problem of improving coordination between an agent and people by means of shared observations. The addition of communication only increases the challenge, making the adaptation of their model far from straightforward.

Another related approach is human-aware planning. The methods of human-aware approach are designed for robot that are meant to work in background. In these cases it is assumed that the humans' agenda (tasks) is independent of the task of the robot and has a higher priority. Therefore the robot is not supposed to influence these plans. For example, Cirillo *et al.* (2010; 2012) describe an agent that generates plans that take into account the expected actions of humans. Tipaldi *et al.* (2011) use spatial Poisson process to predict the probability of encountering humans. While human-aware approaches adjust to human behavior they do not consider their ability to effect that behavior. Moreover, in our settings the robot has private information which is relevant to the success of both it and its human counterpart.

With respect to DEC-POMDPs, over the past decade several algorithms have been proposed to solve them. The traditional DEC-POMDP (Bernstein et al. 2002) models an environment where team members cannot communicate with each other. Solving DEC-POMDP is an NEXP inapproximable problem, thus some researchers have suggested different methods for finding optimal solutions (Szer, Charpillet, and Zilberstein 2005), while others have tried to arrive at the solution using value iteration (Pineau, Gordon, and Thrun 2003; Bernstein, Hansen, and Zilberstein 2005). Several other approaches propose using dynamic programming to find approximated solutions (Szer and Charpillet 2006; Seuken and Zilberstein 2007).

In recent years, a line of work has been suggested which incorporates communication between the teammates. For example, Roth *et al.* (2006) described a heuristic approach for minimizing the number of observations sent if the agent chooses to communicate. They present a DEC-COMM-SELECTIVE (*DCS*) strategy which calculates the best joint-action based on the information known to all team members (observations communicated by team members and common knowledge). The agent then follows the assumption that the other team members will follow the same strategy. This approach ensures coordination when all team members use the same strategy. However, in cases where the agent's teammates do not follow the same strategy, the actions chosen by them may conflict with the actions which the agent considers optimal. Our agent takes this into consideration, and based on a model of its counterpart, tries to coordinate its actions with the predicted actions of its counterpart.

Problem Description

We consider the problem of efficient coordination with communication costs between people and intelligent computer agents in DEC-POMDPs. We begin with a description of the general problem and continue with details of the domain we used to evaluate our agent.

Coordination with Communication Costs

A DEC-POMDP (Bernstein et al. 2002) models a situation where a team of agents (not necessarily computerized ones) has a joint reward (the same goals), and each member of the team has partial observations of the state of the world. The model separates the resolution of the problem into time steps in which the agents choose actions simultaneously. These actions can have deterministic or non-deterministic effects on the state. Following these actions, each team member privately receives an additional observation of the world state. The state transition and the joint reward function are dependent on the joint actions of all agents. In most cases, the reward function cannot be factorized to independent functions over the actions of each agent (such as the sum of rewards for each action). Therefore, the team members must reason about the actions of other teammates in order to maximize their joint rewards. Formally, the model can be described as a tuple $\langle \alpha, S, \{A_i\}, T, \{\Omega_i\}, O, R, \gamma, \Sigma \rangle$, where α denotes the team's size (in our settings $\alpha = 2$), S denotes the set of all distinct world states, A_i is the set of all possible actions that agent i can take during a time step (note that all states and transitions are independent of time) such that A is the set of all possible joint actions, that is $A_1 \times \cdots \times A_\alpha$. T is the transition function $T : S \times A \times S \rightarrow \Re$, which specifies the probability of reaching a state based on the previous

state and the joint-action. Ω_i denotes the possible observations that agent i can receive in a single time step, such that Ω is all possible joint observations $\Omega_1 \times \cdots \times \Omega_\alpha$. O is the observation function $O : S \times A \times \Omega \rightarrow \Re$ which specifies the probability of obtaining a joint observation given the preceding state and the joint-action. Let R denote the reward function $R : S \times A \rightarrow \Re$ based on a joint action in a given state. Finally, γ is the discount factor applied at every time step, that is, given, $s_1, s_2 \in S$, $a \in A$ the actual reward at a given time step t is $\gamma^t \cdot R(s_1, a, s_2)$.

As we allow communication capabilities, we also define Σ as the alphabet of the message so that each $\sigma \in \Sigma$ is a type of observation and $\Sigma^* = \bigcup \Omega_i \cup \{\epsilon\}$. Since the communication incurs a cost, we also use C_Σ to denote the cost function for sending a message $C_\Sigma : \Sigma^* \rightarrow \Re$.

In addition, we use the notion of *belief*, which represents the probability of each state being the correct world state according to the agent's belief. Formally, a belief b is a probability distribution vector such that for each $s \in S$, $b(s)$ is the probability that s is the correct world state and $\sum_{s \in S} b(s) = 1$.

We focus on POMDPs in which the team consists of two agents ($\alpha = 2$) which are able to communicate with each other (e.g., (Roth, Simmons, and Veloso 2006)). As communication is costly, we limit the communication messages to include only self observations. This can also be supported in real settings where limitations occur to prevent sharing additional information which can breach the integrity of the team members (e.g., surrendering their locations). By sharing their observations, the team members can avoid uncoordinated actions caused by contradictory private knowledge, allowing them to build a coherent and concise view of the world states faster.

A naïve approach for team communication is sharing all information among team members. Then, finding the optimal joint action becomes a simple POMDP problem that each team member can solve in parallel. However this solution is only optimal if two assumptions hold. First, that there is no cost associated with communication. Second, that all team members consider the same joint actions to be optimal (by using the same POMDP policy). As this is hardly the case in real settings, existing agents might fail when matched with people. Our agent's design takes these considerations into account in order to achieve proficient interaction with people.

Serbia/Bosnia Domain

To validate the efficacy of our agent, we chose the Serbia/Bosnia domain, which was first introduced by Roth *et al.* (2006)[1] and offered as a benchmark for evaluation of communication heuristics in multi-agent POMDPs. In this domain, two paratroopers are situated in one of two possible 5×5 grid worlds. Each world has a different goal square – $(5, 5)$ or $(2, 4)$ – which represents their extraction point, depending on whether they are located in Serbia or Bosnia, respectively. Each of the team members is aware of the location of the other member in the grid. Yet they do not know in which world grid they are located (be it Serbia or Bosnia). In each time step each member can move either *north*, *south*, *east* or *west*. The agent can also choose to *stop* or send a *signal*. If both team members choose to signal in the correct goal position (that is, the goal square in the world grid in which they are located) at the *same* time step, the team is given a reward of 120 points. If only one team member sends a signal, both signal while in different grid squares or both signal in the wrong goal square, they receive a penalty of -120 points. Regardless of the position, as soon as at least one team member signals, the game ends.

As the agents move they can observe their surroundings (which is saved as private information), thus obtaining new private observations that can help increase their certainty with respect to the correct world grid in which they are situated. The information obtained is one of four types of landscapes: *plain, forest, lake* and *waterfall*. Although all four landscapes exist in both states, Bosnia is characterized with more water landscapes than Serbia, therefore agents are more likely to see a *lake* or a *waterfall* in Bosnia. In Serbia, on the other hand, an agent is more likely to see a *plain* or a *forest*. The probability of seeing each landscape depends only on whether the team is in Serbia or Bosnia, and not on the current grid position in which the agent is located. Each team member can share its observations (e.g., "forest") with a given communication cost of -2. Sharing information can help the team reach a swift conclusion about the current world. Due to restrictions, applied also in real settings (such as security domains or military operations), the communication is restricted solely to observations, thereby prohibiting the exchange of strategy related information or decisions. Each movement also costs -2. In addition, a discount factor of $\gamma = 0.9$ exists, whereby the rewards and penalties decrease as time progresses. Note that in this domain the decision that has the highest immediate effect on the reward is whether or not to signal.

Agent Design

As we demonstrate later, the current automated state-of-the-art agent teamed with people achieved poor coordination. The main reason for this is the inherent behavior of people. People tend to make mistakes as they are affected by cognitive, social and cultural factors, etc. (Lax and Sebenius 1992). Moreover, it has been shown that people do not follow equilibrium strategies (Erev and Roth 1998), nor do they maximize their expected monetary values. This behavior, if unaccounted for, might have undesirable effects on the strategy of agents interacting with people.

When coordinating with someone else, it is hard to predict what the other team member (especially if it is a human partner) will do. The task is even harder if the agent interacts with someone only once and not repeatedly. Thus, an efficient agent working with people needs, amongst other things, to approximate what percentage of the population will perform each action based on the existing partial observations. Our agent interacts with the same counterpart only once and thus its design tries to tackle the challenge by generating a good model of the population based on an existing

[1] We used slightly different titles and parameters in our experiments for the sake of simplicity.

knowledge base. By doing so, it also considers people's deviation from the policy that maximizes the monetary value. This allows the agent to maximize the average score for the entire team. Since the agent builds a good model of the team and works in decentralized communication settings, we coin it *TMDC* (team modeling with decentralized communication).

Modeling People's Behavior

We believe that an efficient coordination of agents in mixed agent-human environment requires a good model of people's behavior. To achieve this we gathered information, using the Amazon Mechanical Turk framework, about people's behavior in the domain wherein our agent is situated. As our domain requires only a short interaction between team members our human behavior's model was developed accordingly.

First we matched people with automated agents to gather a set of decisions made by people in different settings of the domain. After having a substantial amount of data, we used a machine learning technique. Based on the domain, we chose which features of the actions and state of the world are relevant for the learning. We used a neural network model to estimate the distribution of people's behavior, whereas the input to the network consisted of the different features and the output consisted of the different feasible actions. The model is then used to obtain a probability measure with respect to the likelihood of the human player to choose a given action in a given setting of the domain.

We had used a large knowledge base of the decisions made by more than 445 people who played the game. For each decision we generated a set of features which included the position, belief, last communicated observations and last actions of each team member. These features were used as the input for the neural network model. We learned a neural network using a genetic algorithm with $1/MSE$ as the fitness function. The output of the model was normalized to 1 and was treated as the probability that the human partner will take each action. As people make decisions using private information that the automated agent is unaware of, our model's features try to "estimate" what observations people actually had and thus what is their belief with respect to the state of the world.

In order to improve the precision of the model, we separated the data samples into three sets based on positions, and grouped together the outputs of equivalent actions. Then we used the model to return a probability vector indicating the likelihood that the human player will choose each of the 6 actions defined in our domain.

The neural network model had 13 inputs, 3 outputs and 8 neurons in its hidden layer. The input features included four beliefs generated on four sets of observations: (a) all observations sent by both team members, (b) all observations sent by the agent, (c) all observations sent by the human player and (d) all observations known to the agent. Two more features encode the last shared observation of each team member and another feature is the last observation shared by any player. Two additional features represent the direction of each player's last movement. The last four features encode position related information – which player is closer to each goal and whether a player is already in it. The mean square error of the model was 0.16 with a precision of 63.5%.

Designing the Agent's Strategy

The general design of the agent's strategy consists of building a POMDP using the prediction of the human behavior described beforehand. Thus, *TMDC* uses its model, and not the shared belief, to predict what its counterpart's behavior will be. In addition, *TMDC* chooses its action based on all of its knowledge (which also includes private knowledge), and only communicates in order to influence the actions of the other teammate. Given all previously shared observations, the agent evaluates an action by considering all possible results, calculating immediate rewards and using offline estimation of future rewards. This evaluation is then used by a hill climbing heuristic that finds which observations (taken from the set of all observations, including shared observations) can maximize the score of the team and hence should be shared.

Let $A_1 \subseteq A$ be the set of actions available to the agent and $A_2 \subseteq A$ be the counterpart's possible actions. Let H^t be all indicators (past actions, communicated observations and team's position on the grid) the agent has of its partner's behavior at time step t. Let M be the prediction function, which, based on H^t, specifies the probability of it choosing a specific action. Let b^t be the agent's belief based on all shared observations and its private observations, and V be the estimated value of a given belief and history, described hereafter. We then formally define the agent's score of an action, where the Q function employs a strategy of a 1-step look ahead:

$$\begin{aligned} Q(b^t, H^t, a_1) &= \\ \sum_{a_2 \in A_2} M(H^t, a_2) &\cdot (\sum_{s \in S} b^t(s) \cdot R(s, (a_1, a_2)) \\ +\gamma \sum_{\omega \in \Omega_1} Pr(\omega|(a_1, a_2), b^t) &\cdot V(b^{t+1}, H^{t+1})) \end{aligned} \quad (1)$$

The agent calculates the action based on the predicted distribution of the rest of the team choosing each action, and estimates future utility, based on possible future beliefs. These beliefs are effected by both the actions and the possible next observations. The history is also updated, adding the new actions and their effect on the common knowledge (positions in the grid). The updated belief functions and the probability of obtaining an observation given an action are calculated as follows:

$$b^{t+1}(s\prime) = \frac{O(s\prime, (a_1, a_2), \omega) \sum_{s \in S} T(s, (a_1, a_2), s\prime) b^t(s)}{Pr(\omega|(a_1, a_2), b^t)} \quad (2)$$

$$\begin{aligned} Pr(\omega|(a_1, a_2), b^t) &= \sum_{s\prime \in S} O(s', (a_1, a_2), \omega) \cdot \\ &\sum_{s \in S} T(s, (a_1, a_2), s\prime) b^t(s) \end{aligned} \quad (3)$$

Based on the Q function, the agent employs a hill climbing heuristic to search for the optimal message and the optimal action for that message. The agent first calculates the optimal actions, assuming the message will either be empty

or contain all its observations, denoted a_{nc} and a_c, respectively. This is somewhat similar to the approach used by Roth *et al.* (2006). The agent then searches for the optimal message for each a_{nc} and a_c by repeatedly adding observations to the outgoing communication a_{nc} that increases the expected score of the action. The algorithm will finally send the message that achieves the highest expected score while taking communication costs into consideration.

We are now left to define the value of a belief $V(b^t, H^t)$. Perhaps the most time-efficient approach to approximate future rewards is evaluating the optimal score that can be achieved by the team in each state if the true state would be revealed to all players. This approach was also used to solve POMDPs after a 1-step look ahead, and used in *DCS* after two steps. However, it is well documented that this approach does not give accurate approximations and gives preference to delaying actions (Littman, Cassandra, and Kaelbling 2005). Thus a different approach is needed. Another simple approach is to use value iteration to evaluate the score of an MDP where every (belief, history) is a state. Unfortunately, such an MDP has an infinite number of states, as both the belief and possible histories have infinite value ranges. The agent therefore creates an abstract model with a reasonable number of states, by creating discrete and compact representations, as described hereafter. The abstract model is compact and consists of only a subset of fields derived from the game's history. The agent also creates discrete resolutions for the continuous fields. The model's states represent the positions of the team members (since we have a 5×5 grid, we have 25 possible position values for each team member), the last actions taken by each team member (categorized according to 3 possible values: moving towards Serbian goal, Bosnian goal or no movement), the private belief of the agent (using 17 discrete values) and the shared belief derived from the communication history (using 17 discrete values). Thus, this model has $1,625,625$ possible states ($25 \times 25 \times 3 \times 3 \times 17 \times 17$). The agent then uses the following update function for value iteration:

$$
\begin{aligned}
V_n(b^t, b_s^t, a_{1,t-1}, a_{2,t-1}) \quad &= \quad max\Big(\max_{a_1 \in A_1} \sum_{a_2 \in A'} M(H^t, a_2) \\
\cdot\Big(\sum_{s \in S} b^t(s) \cdot R(s,(a_1,a_2)) \quad &+ \quad \gamma \sum_{\omega \in \Omega_1} Pr(\omega|(a_1,a_2), b^t) \\
\cdot V_{n-1}(b^{t+1}, b_s^t, a_1, a_2)\Big), \quad &\max_{b'_s \in resolution} V_{n-1}(b^t, b'_s, a_{1,t-1}, a_{2,t-1}) \\
-CommunicationCost\Big)
\end{aligned}
\tag{4}
$$

where b^{t+1} is an updated belief previously defined, b_s^t is the shared belief, $a_{i,t-1}$ are the previous actions, H^t are the fields required for the predicted model synthesized from the available fields and $CommunicationCost$ is a general estimated cost to change the shared belief from b_s^t to b'_s (following a pessimistic assumption about the weakest observations).

This value iteration update function converges after approximately 40 iterations calculated once *offline*. Thus, when evaluating an action's score, the agent uses the approximated value of the abstract state with the nearest discrete values for the shared and private beliefs.

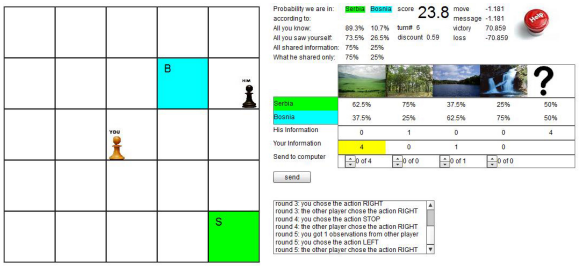

Figure 1: The game interface used in the experiments.

Experiments

The experiments were conducted on the Bosnia/Serbia domain using the Amazon Mechanical Turk service (AMT)[2]. This framework allows the publishing of tasks designated for people all around the world. We prohibited multiple participation by the same people. We begin by describing the experimental methodology and then continue by presenting the experimental results.

Experimental Methodology

The players were shown a presentation explaining the game and their tasks before their participation. Although the presentation is very detailed, we took great care not to give strategic advice. We then required that each worker pass a short multiple choice test to verify that they read the manual and understood the game. Each player who completed the game received a minimal payment of ¢30. To motivate the players to play seriously and be focused on the game, each player received a bonus equivalent to the number of cents based on the team's score, if it was positive. We set the starting score of the game to 40 to ensure that the costs and penalties of the game would have a meaningful effect on the player even if the team did not gain the reward for a successful signal.

As for the game's interface, at every time step the players were shown the current value of movement and communication and a successful/failed signal. The interface also displayed to the players the number of observations seen, received and sent, as well as the probability of each grid state, based on Bayes' rule. A screen-shot of the interface is shown in Figure 1. We selected four pairs of starting positions at random, in advance, for the game settings. We created two scenarios for each of the starting positions, one with Serbia as the true grid state and the other with Bosnia. An equal number of games in each scenario were run for each agent.

We provided four belief probability values to the player, based on different available observations and beliefs. However, it is up to the human player to take these probabilities into account. The four beliefs are generated from subsets of observations available to the player: all observations known to the player, observations seen by the player herself, observations shared by the other player and observations shared by the agent.

[2]For a comparison between AMT and other recruitment methods see (Paolacci, Chandler, and Ipeirotis 2010).

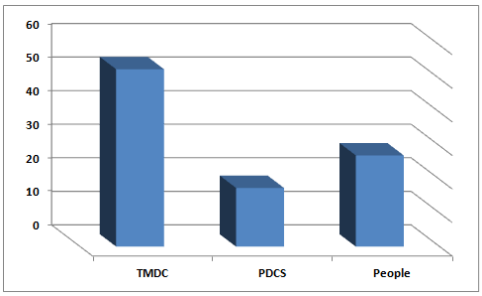

Figure 2: Average scores obtained by each team type.

We experimented and compared our agent with an agent based on the state-of-the-art *DCS* strategy (Roth, Simmons, and Veloso 2006), namely a polynomial version of DEC-COMM-SELECTIVE (*PDCS*). This agent finds the joint-actions a_{nc} with a maximal score based on a belief generated by the shared observations. It then finds the joint-actions a_c based on all its observations. The agent then creates the minimal message required to convince the other player, based on shared belief, that a_c is the best joint-actions. The agent only communicates its observations if the score difference between a_c and a_{nc} is greater than the communication costs. Basically, the *PDCS* agent expects the team to perform the optimal plan based on common knowledge. It updates the common knowledge when it believes changing the plan warrants the communication cost.

Experiment Results

We matched 64 human players with each agent (*TMDC* and *PDCS*) and paired 128 human players with each other. This section analyzes the results obtained by the agents as well as providing an in-depth analysis of human behavior.

Evaluating Agents' Strategies Figure 2 summarizes the scores obtained by each team type. The results demonstrate that our agent significantly outperforms the *PDCS* agent ($p < 0.0001$) when paired with people. The average score for *TMDC* was 52.84, compared to only 17.5 obtained by the state-of-the-art *PDCS* agent. The pure human-human teams achieved an average score of 27.18. While the difference between the scores of pure human-human teams and the *PDCS*-human teams were not significant, the *TMDC*-human teams achieved significantly higher results ($p = 0.003$) from the human-human teams as well.

We also tested how well the *PDCS* agent coordinates with itself. It achieved a score of 65.8 (in 400 games). It is not surprising that the *PDCS*-*PDCS* team outperformed *TMDC*-human teams. The *PDCS* agent can fully predict and coordinate with itself, while a human partner is not fully predictable and may employ inefficient communication and action policies. In fact, as *PDCS* is a state-of-the-art multi-agent coordination algorithm, its results are near-optimal. It is, however, interesting to note that the results of the *TMDC*-human teams are closer to the results of a *PDCS*-*PDCS* team than to that of the human-human teams.

The results demonstrate the success of incorporating a prediction model in the inherent design of the agent domain. For instance, it allowed our agent to gain advantage and essentially allowed it to wait outside the goal until it believed signaling was an optimal action. The *PDCS* agent assumes that its partners will not signal until the shared information indicates that signaling is optimal. Therefore, the agent may enter the goal square immediately, which can result in uncoordinated signals and a low score for the team. As human players make different decisions they can also make different mistakes. For example, some may choose to wait even if their observations are very conclusive, while others may try to reach a goal quickly and signal even if they do not have sufficient evidence or whether they are even in the presence of contradicting evidence.

Agent	Observations Sent by Agent	Observations Sent by People
TMDC	1.25	2.72
PDCS	1.97	2.48
People	N/A	2.86

Table 1: Average number of observations shared by each player

Table 1 summarizes the number of observations sent by the team members. A human player sends 2.84 observations on average per game, significantly more than the *PDCS* agent, which sends only 1.97 observations. While the *PDCS* communication policy considers one observation to be sufficient motivation to more toward a specific goal and two additional observations to motivate a signal, the *TMDC* agent communicates significantly less than both human players and the *PDCS* agent, sending only 1.25 observations on average each game. The reason for that is the fact that the *PDCS* agent sends more observations based on supporting or contradictory observations sent by the human player and based on the observation's quality (e.g., being *forest* or *plain*). The *TMDC* agent, on the other hand, takes into account that sending only a single observation influences only a subset of the population and not all of it, and that sending additional observations can increase the proportion of the population that will be convinced to move in the direction the agent believes to be the right one. Thus sending additional observations becomes a tradeoff between the cost of communication and the score gained by increasing the probability that the human player will make the correct move.

Conclusions

Settings in which hybrid teams of people and automated agents need to achieve a common goal are becoming more common in today's reality. Communication in such situations is a key issue for coordinating actions. As communication is costly and sometimes even limited (e.g., due to security issues or range limitations), it becomes of great essence to devise an efficient strategy to utilize communication. This paper presented a novel agent design that can proficiently coordinate with people under uncertainty while taking into account the cost of communication.

Our agent was specifically designed taking into account the fact that it interacts with people, and it was actually evaluated with people. The success of our agent's proficiency with people cannot be overstated. Experiments with more than 200 people demonstrated that it outperforms a state-

of-the-art agent and even people. One of the main factors accounting for the success of our agent is the understanding that it requires a good model of the counterpart to generate an efficient strategy.

Though the Serbia/Bosnia domain that we used was quite a compact one and only included uncertainty on a single issue (the country in which the agents are located), we found that it was hard for human team members to incorporate this information into their strategy. We believe that the lack of correlation between choosing to signal and the probability of being in the correct goal is partially caused by the probabilistic nature of the information. Our hypothesis is that human players will pay more attention to observations if the observations give concrete definitive information. Nevertheless, regardless of this non-efficient behavior of people, once our agent builds the model it can efficiently coordinate with them and generate higher rewards for the team. Future work will also situate our agent in domains where observations would not only change the likelihood of states but will allow eliminating possible states as well.

This paper is just part of a new and exciting journey. Future work warrants careful investigation on improving the prediction model of people's behavior. We will also investigate settings in which even more limited information is available to the team members. In such situations the challenge is in understanding the abstract model that is available and how to utilize communication for efficient coordination that will allow for the increased accuracy of the model.

References

Bernstein, D.; Givan, R.; Immerman, N.; and Zilberstein, S. 2002. The complexity of decentralized control of markov decision processes. *Mathematics of Operations Research* 27(4):819–840.

Bernstein, D. S.; Hansen, E. A.; and Zilberstein, S. 2005. Bounded policy iteration for decentralized POMDPs. *IJCAI*.

Bradshaw, J. M.; Sierhuis, M.; Acquisti, A.; Feltovich, P.; R. Hoffman, R. J.; Prescott, D.; Suri, N.; Uszok, A.; and Hoof, R. V. 2003. *Agent Autonomy*. Dordrecht, The Netherlands: Kluwer. chapter Adjustable autonomy and human-agent teamwork in practice: An interim report on space applications, 243–280.

Bradshaw, J.; Feltovich, P.; Johnson, M.; Bunch, L.; Breedy, M.; Eskridge, T.; Hyuckchul, J.; Lott, J.; and Uszok, A. 2008. Coordination in human-agent-robot teamwork. In *CTS*, 467–476.

Breazeal, C.; Kidd, C.; Thomaz, A.; Hoffman, G.; and Berlin, M. 2008. Effects of nonverbal communication on efficiency and robustness in human-robot teamwork. *IROS*.

Broz, F.; Nourbakhsh, I.; and Simmons, R. 2008. Planning for human-robot interaction using time-state aggregated POMDPs. *AAAI* 102–108.

Casper, J., and Murphy, R. R. 2003. Human-robot interactions during the robot-assisted urban search and rescue response at the world trade center. *IEEE Transactions on Systems, Man & Cybernetics: Part B: Cybernetics* 33(3):367 – 385.

Cirillo, M.; Karlsson, L.; and Saffiotti, A. 2010. Human-aware task planning: an application to mobile robots. *TIST* 15:1 – 15:25.

Cirillo, M.; Karlsson, L.; and Saffiotti, A. 2012. Human-aware planning for robots embedded in ambient ecologies. *Pervasive and Mobile Computing*. to appear.

Doshi, P., and Gmytrasiewicz, P. J. 2009. Monte carlo sampling methods for approximating interactive POMDPs. *Artificial Intelligence Research* 34:297 – 337.

Erev, I., and Roth, A. 1998. Predicting how people play games: Reinforcement learning in experimental games with unique, mixed strategy equilibrium. *American Economic Review* 88(4):848–881.

Kamar, E.; Gal, Y.; and Grosz, B. 2009. Modeling user perception of interaction opportunities for effective teamwork. In *IEEE Conference on Social Computing*.

Lax, D. A., and Sebenius, J. K. 1992. Thinking coalitionally: party arithmetic, process opportunism, and strategic sequencing. In Young, H. P., ed., *Negotiation Analysis*. The University of Michigan Press. 153–193.

Littman, M. L.; Cassandra, A. R.; and Kaelbling, L. P. 2005. Learning policies for partially observable environments: Scaling up. *ICML*.

Paolacci, G.; Chandler, J.; and Ipeirotis, P. G. 2010. Running experiments on Amazon Mechanical Turk. *Judgment and Decision Making* 5(5).

Pineau, J.; Gordon, G.; and Thrun, S. 2003. Point-based value iteration: An anytime algorithm for POMDPs. *IJCAI*.

Rosenthal, S.; Biswas, J.; and Veloso, M. 2010. An effective personal mobile robot agent through symbiotic human-robot interaction. In *AAMAS*, 915–922.

Roth, M.; Simmons, R.; and Veloso, M. 2006. What to communicate? execution-time decision in multi-agent POMDPs. *Distributed Autonomous Robotic Systems*.

Sarne, D., and Grosz, B. J. 2007. Estimating information value in collaborative multi-agent planning systems. In *AAMAS*, 227–234.

Seuken, S., and Zilberstein, S. 2007. Memory-bounded dynamic programming for DEC-POMDPs. *IJCAI*.

Shah, J.; Wiken, J.; Williams, B.; and Breazea, C. 2011. Improved human-robot team performance using chaski, a human-inspired plan execution system. In *HRI*.

Szer, D., and Charpillet, F. 2006. Point-based dynamic programming for DEC-POMDPs. *AAAI*.

Szer, D.; Charpillet, F.; and Zilberstein, S. 2005. Maa*: A heuristic search algorithm for solving decentralized POMDPs. *UAI* 576–583.

Tipaldi, D., and Arras, K. 2011. Please do not disturb! minimum interference coverage for social robots. In *IROS*, 1968 – 1973.

van Wissen, A.; Gal, Y.; Kamphorst, B.; and Dignum, V. 2012. Human–agent teamwork in dynamic environments. *Computers in Human Behavior* 28:23–33.

Woods, D. D.; Tittle, J.; Feil, M.; and Roesler, A. 2004. Envisioning human-robot coordination in future operations. *IEEE Transactions on Systems, Man & Cybernetics: Part C: Special Issue on Human-Robot Interaction* 34:210–218.

Zuckerman, I.; Kraus, S.; and Rosenschein, J. S. 2011. Using focal points learning to improve human-machine tactic coordination. *JAAMAS* 22(2):289–316.